

Colin H. Murray (State Bar No. 159142)

colin.murray@bakermckenzie.com

Teresa H. Michaud (State Bar No. 296329)

teresa.michaud@bakermckenzie.com

BAKER & McKENZIE LLP

Two Embarcadero Center, 11th Floor San Francisco, CA 94111

Telephone: +1 415 576 3000

Facsimile: +1 415 576 3099

W. Barton Rankin (Admitted Pro Hac Vice) w.rankin@bakermckenzie.com

BAKER & McKENZIE LLP

2300 Trammell Crow Center 2001 Ross Avenue

Dallas, Texas 75201

Telephone: +1 214 978 3000

Facsimile: +1 214 978 3099

Attorneys for Plaintiff Lilith Games

**UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN FRANCISCO DIVISION**

LILITH GAMES (SHANGHAI) CO.
LTD.

PLAINTIFF,

VS.

UCOOL, INC. AND UCOOL, LTD.

DEFENDANTS.

Case No. 3:15-cv-01267-SC

**DECLARATION OF KENDYL A.
ROMAN IN SUPPORT OF
PLAINTIFF'S MOTION FOR
PRELIMINARY INJUNCTION**

District Judge Samuel Conti

Date: August 14, 2015

Time: 10:00 a.m.

Place: Courtroom 1, 17th Floor

CONFIDENTIAL – UNDER PROTECTIVE ORDER

TABLE OF CONTENTS

1		
2	TABLE OF CONTENTS	1
3	I. INTRODUCTION	2
4	II. RECENT TESTIMONY	3
5	III. QUALIFICATIONS	4
6	IV. ASSIGNMENT	8
7	V. MATERIALS REVIEWED	8
8	VI. EXHIBITS	9
9	VII. LEGAL STANDARDS APPLIED	9
10	VIII. OPINIONS AND BASES THEREFORE	10
11	A. Summary of Opinions	10
12	IX. TECHNICAL BACKGROUND	10
13	A. Computer Hardware and Software	10
14	B. Object Code and Source Code	11
15	C. Programming Languages	11
16	D. Code Comparison Tools: diff and beyond	11
17	E. Obfuscation: Defeating Diff by Obscuring Code	12
18	X. Overview of the Lilith's Sword and Tower	14
19	XI. Overview of the UCOOL'S Heroes Charge	16
20	XII. ANALYSIS	17
21	A. Analysis Overview	17
22	B. Analysis Results	17
23	XIII. CONCLUSION	29
24		
25		
26		
27		
28		

I. INTRODUCTION

1. I, Kendyl A. Román, have been engaged by Baker & McKenzie LLP, attorneys for plaintiff Lilith Games (“Lilith”), as a computer forensic expert in the field of computer software and hardware, and general knowledge of computer science.

2. The plaintiff in this case is Lilith which is the developer of a mobile game known as 刀塔传奇 in Chinese (see logo below) (Romanized as “Dāo Tǎ Chuánqí”), which is referred to in this case as “Sword and Tower Legend” or as “DOTA Legend” (collectively “Sword and Tower”).



3. The defendants in this case are uCool, Inc. and uCool, Ltd. (collectively, “uCool”).

4. Lilith accuses uCool of copyright infringement and trade secret misappropriation based on uCool’s publication and marketing of the game “Heroes Charge” beginning in or around August of 2014.

5. My services are compensated at the rate of \$525.00 per hour. The services of my staff are compensated at a lower rate. No part of my compensation depends in any way on the outcome of this litigation.

1 **II. RECENT TESTIMONY**

2 6. Cases in which I have testified as an expert witness at trial or by
3 deposition during the previous five years are identified as:

- 4 • Eastern District of North Carolina, *SAS Institute Inc. v. World*
5 *Programming Limited*, 5:10-cv-00025
6 • Central District of California, Los Angeles, *MyKey Technology Inc.*
7 *Patent Litigation*, 2:13-ml-02461
8 • District of Delaware, *Parallel Networks v. A10 Networks*, 13-cv-1943,
9 and *Parallel Networks v. F5 Networks*, 13-cv-2001
10 • U.S. Patent Trial and Appeal Board, *TD Ameritrade v. Trading*
11 *Technologies International*, CBM2014-00131, 133, 135, and 137
12 • Southern District of New York, *Sanford and DYMO v. Esselte*, 1:14-cv-
13 07616
14 • Northern District of Ohio, *Hickok Inc. v. SysTech International, LLC*,
15 1:07-cv-03565
16 • Eastern District of Virginia, Alexandria, *Reporting Technologies, Inc. v.*
17 *Emma, Inc.*, 1:11-cv-01203
18 • Central District of California, Los Angeles, *Futurelogic, Inc. v. Nanoptix,*
19 *Inc.*, 2:10-cv-07678
20 • Southern District of Texas, Houston, *Wellogix v. Accenture*, 3:08-cv-119
21 • Northern District of California, San Jose, *Embry v. Acer America*, 09-cv-
22 01808
23 • Southern District of Texas, Houston, *Wellogix v. BP*, 4:09-cv-1511

24 7. Cases where I have provided recent reports or declarations but
25 have not testified:

- 26 • Central District of California, Los Angeles, *Miller vs. Fuhu*, 14-cv-6119
27
28

- 1 • U.S. Patent Trial and Appeal Board, *GoPro, Inc. v. Contour, LLC*,
2 IPR2015-_____
- 3 • District of Minnesota, *Twin City Fan Companies, Ltd. v. FPT Software*,
4 12-cv-1357
- 5 • Superior Court of California, Riverside County, Indio, *Malanche v.*
6 *Eisenhower Medical Center, INC* 1108128
- 7 • Central District of California, Western Division, *Innersvingen AS v.*
8 *Sports Hoop, Inc.*, 2:12-cv-05257
- 9 • Western District of Oklahoma, *Paycom Payroll, LLC v. Richison and*
10 *Period Financial Corporation*, 09-cv-488

11 **III. QUALIFICATIONS**

12 8. See my Curriculum Vitae, attached as Roman Exhibit 1, for a
13 listing of my qualifications. This includes a list of publications for the past 10
14 years or more.

15 9. My expertise qualifies me to do the type of analysis required in
16 this case. Of particular relevance, I have been involved in the design,
17 implementation, testing, and analysis of computer software and hardware for over
18 thirty five years, including developing multimedia and game authoring tools and
19 frameworks and analyzing large computer source code codebases for copyright
20 infringement and trade secret misappropriation. In addition, I have practical
21 experience in the design and programming of a variety of computer systems
22 ranging from laptops and desktop computers to large multi-layer networked
23 database systems.

24 10. Over the past sixteen years, I have performed forensic analysis of
25 computer software in dozens of cases including, as a court appointed expert in San
26 Jose, CA, *Aspect Communications Corporation v. eConvergent, Inc. et al.* and
27
28

1 Ribeiro v. Weichselbaumer, and as a Federal Special Master in *Paycom Payroll,*
 2 *LLC v. Richison and Period Financial Corporation.*

3 11. In *Aspect Communications Corporation v. eConvergent, Inc. et al.*
 4 I conducted a technical hearing where both parties presented their expert opinions
 5 and attorney arguments regarding the specificity of the trade secret disclosures and
 6 other discovery issues. I then submitted a technical opinion which was adopted by
 7 the court in its legal opinion.
 8

9 12. In *Paycom Payroll, LLC v. Richison and Period Financial*
 10 *Corporation*, as Special Master, I performed copyright infringement analysis of a
 11 second alleged instance of infringement of a source code copyright. The court
 12 adopted my opinion.

13 13. As a freshman at Brigham Young University (“BYU”) in 1976, I
 14 started writing programs for IBM computers.

15 14. In 1980, I worked with Apple II computers and wrote the first
 16 games with graphical user interfaces. Also at BYU, I was elected President of the
 17 BYU Chapter of the Association for Computing Machinery (“ACM”).¹

18 15. I graduated with High Honors from Brigham Young University
 19 where I received a Bachelor of Science degree in Computer Science. My formal
 20 studies included computer programming, programming languages, algorithms,
 21 operating systems, database systems, digital logic design, and computer
 22 architecture.
 23

24 16. In 1981, I worked at International Business Machines (“IBM”) in
 25 San Jose, CA. During my employment at IBM, the IBM PC was released.
 26
 27

28 ¹ I am a Professional Member of the ACM. I am also a member of the Institute of Electrical and
 Electronics Engineers, Inc. (“IEEE”).

1 17. In 1982, at Dialogic, I improved the performance of the Computer
2 Aided Design (“CAD”) software² and developed a new language called Diagnostic
3 Command Language (“DCL”) which was an interpreted language (like Java and
4 Lua).

5 18. Around 1984, I became a Certified Apple Developer. The first
6 product I developed and marketed was MacBaby® Math, which was educational
7 software designed for very young children.³

8 19. I started developing multimedia animations in 1989.

9 20. In 1990, I started working with digital video. In particular I
10 developed Apple’s Macintosh Programming Fundamentals Self-Paced Training
11 Course on CD-ROM.

12 21. In 1991 and 1992, I worked on the Newton team at Apple and for
13 Slate. The Newton was the commercially available Personal Digital Assistant
14 (“PDA”), a handheld computing device. Slate made software for early tablet
15 computers, such as the NCR tablet. I owned and used both a Newton PDA and a
16 NCR tablet.

17 22. In the early 1990s, I formed the Multimedia Lab at The Carl
18 Group. Projects included developing graphic animations, developing multimedia
19 authoring tools, and various interactive CD-ROM titles. My work with multimedia
20 authoring tools included developing low-level software for both the Macintosh and
21 IBM PC platforms.

22 23. I analyzed and wrote the manual for the Apple Media Tool
23 Programming Environment. The Apple Media Tool allowed developers to create
24
25

26
27 ² The software, the Lucas Drawing System, had been developed by LucasFilms to aid in the
production Star Wars.

28 ³ MacBaby® Math was shown in MacWorld magazine in 1987 and was reviewed by the Boston
Computer Society in September 1990. See www.wolfpup.org/misc/MacBaby_Math_review.pdf.

1 multimedia applications, including games that could run on multiple platforms, i.e.
2 Macintosh and Multimedia PCs. Apple Media Language was an object oriented
3 programming language.

4 24. At The Carl Group, we developed extensions to the Apple Media
5 Language and its framework. Our product, XMeDIA, was used by many
6 multimedia developers including game developers.

7 25. I also worked with the Apple SK8 Team that was developing its
8 own multimedia authoring language and framework. One subproject was called
9 Coco, which empowered children to do simple programming.

10 26. My Curriculum Vitae identifies over 65 issued U.S. and
11 International patents and over 65 published patent applications for which I am
12 listed as an inventor or assignee. For example, in one of my patent applications,
13 filed in 2007, I disclosed a wireless handheld device with a touch screen and
14 audio/video capabilities, include streaming video. Further for example, US Patent
15 8,282,493 and 8,500,563 claim games that run, for example, on an iPhone or other
16 mobile device.

17 27. Prior to being retained in this matter, I have acquired and
18 performed forensic analysis of several computer systems. In *ITC 337-TA-743*
19 (*Motiva v. Nintendo Co Ltd, Nintendo of America Inc*), I analyzed the system
20 software and game software of the Nintendo Wii. In *Electro Source, LLC. v. Fire*
21 *Intl' Ltd. Inc, et al.* I analyzed various PlayStation 2 games and their cheat codes.
22 As a result, I have had access to the type of components and information at issue in
23 this case and have contemporaneous knowledge of what was publicly known.

24 28. Both Federal and *State* Courts have recognized me as an expert in
25 computer architecture, computer software, database systems, networks, and
26 computer forensic science.
27
28

29. For example, a Texas District Court and the Fifth Circuit have recognized me as a computer forensic expert,⁴ in a case involving analysis of software, including source code, in a trade secret case. The U.S. Supreme Court did not find it necessary to further review my qualifications and testimony.⁵

30. In general, the subject matter of my work in these cases involved intellectual property related to computer software structure and operation. The technology in this case is computer software for mobile games.

IV. ASSIGNMENT

31. I have been asked to express an opinion regarding the copyright infringement and trade secret misappropriation associated with Lilith's Sword and Tower by Defendants.

V. MATERIALS REVIEWED

32. In my study of this matter, I have had access to the materials as set forth in Exhibit 2 and considered and relied on these materials to varying degrees. Citations to these materials that appear below are meant to be exemplary but not exhaustive. I have also relied on my professional knowledge and experience.

33. I have analyzed the Lilith and uCool source code as produced. I also analyzed the gaming frameworks of Cocos and Unity.

34. I understand that certain aspects of discovery in this case have not been completed. There are items that have not been produced or reviewed by me. I reserve the right to consider and comment on any additional or different materials provided by parties or the Court, including providing additional reports or opinions.

⁴ See Wellogix, Inc. v. Accenture, LLP, No. 11-20816, United States Court of Appeals, Fifth Circuit, May 15, 2013.

⁵ See No. 13-1051, United States Supreme Court where such a petition was denied.

1 35. Accordingly, my investigation and review in this matter is
2 continuing and this declaration represents my findings based on my review of
3 information available as of the date of this report. Further, I reserve the right to
4 consider and comment on any additional expert statements and testimony provided
5 by fact witnesses or other experts in this matter. Any changes or necessary
6 modification of these conclusions will be addressed in supplemental reports or
7 testimony, which I reserve the right to provide. I also reserve the right to rely on
8 demonstrative exhibits to supplement my testimony at the hearing.
9

10 **VI. EXHIBITS**

11 36. Roman Exhibit 1 is my *Curriculum Vitae*.

12 37. Roman Exhibit 2 lists the materials reviewed, considered or made
13 available to me.

14 38. Roman Exhibit 3 presents graphics from the software.

15 39. Roman Exhibit 4 presents files lists.

16 40. Roman Exhibit 5 source code analysis.

17 41. The remaining Exhibits are documents produced in this case and
18 referenced herein.

19 **VII. LEGAL STANDARDS APPLIED**

20 42. I am not a lawyer and will not provide any legal opinions.

21 Although I am not a lawyer, I have been advised certain legal standards are to be
22 applied by technical experts in forming opinions. In particular, I understand that
23 the principles that govern expert witness analysis and evidence are set forth in Rule
24 702, F.R.E. and in *Daubert v. Merrell Dow Pharmaceuticals, Inc.*, 509 U.S. 579
25 (1993), and *Kumho Tire Co. v. Carmichael*, 119 S.Ct. 1167 (1999).
26
27
28

VIII. OPINIONS AND BASES THEREFORE

A. Summary of Opinions

43. Based on my review thus far of the software, documents and computer-based data, materials, and testimony in the case, the following is a summary of my opinions:

- a. The details of the locales, settings, characters, themes, plot, sequence of play, staging, playing of music and sound, and tutorial from Sword and Tower are specifically embodied in the Lua source code or data structures defined in the Lua source code.
- b. Based on Mr. Kitchen's findings and confirmed by my own analysis, it is my opinion that uCool had access to Litith's source code.
- c. Significant portions of the software program were literally copied and are identical.
- d. Significant portions of the software program were literally copied and then translated, modified, reordered, or otherwise obscured in such a way that common code comparison programs would not be able to detect the similarities.

IX. TECHNICAL BACKGROUND

44. The following sections contain detailed technical background covering to the technology and issues in this case.

A. Computer Hardware and Software

45. Originally, computers were composed only of physical circuits, known as hardware,⁶ that were programmed by physically configuring wires (like a telephone switch board operator). Computer programs (a series of computer

⁶ Hardware is the tangible components of a computing system, such as vacuum tubes, wires, circuit boards and other discrete components.

instructions) stored in memory are known as software,⁷ because they can be modified much more easily than hardware. In the late 1970s, personal computers (PCs) became commercial products (such as the Apple II in 1977).

B. Object Code and Source Code

46. Typically, the computer software program executes on the computer hardware in a form that is easy for the machine to read; this is referred to as machine code or object code. The object code is typically compiled or interpreted from a human readable version of the software program. The human readable (and writeable) version of the software program is known as source code.

C. Programming Languages

47. The source code instructions that make up a computer program may be written in one of more programming languages. Some languages like the “C” Programming Language are compiled into object code from the C source code. Other languages like Java are interpreted languages where the instructions are processed and turned into machine level instructions when the program is execution. Some interpreted languages are interpreted into an intermediate form called byte code.

48. C++ is an objected oriented version of C.

49. C# or C-sharp is Microsoft’s creation of a language that is like Java and C++.

D. Code Comparison Tools: diff and beyond

50. The diff utility was developed by AT&T Bell Laboratories with Unix. The final version, first shipped in 1974. A research paper regarding the

⁷ Software is a “generic term for those components of a computer system that are intangible rather than physical. It is most commonly used to refer to the programs executed by a computer system as distinct from the physical hardware of that computer system, and to encompass both symbolic and executable forms for such programs.” (Oxford Dictionary ‘software’)

1 Hunt–McIlroy algorithm was published in 1976.⁸ The diff program is well known
 2 and widely used. I have used it for over thirty years. It is useful to quickly see
 3 how two versions of the same file have been changed. The diff utility has been
 4 made available for Windows and Macintosh as well as many versions of Unix,
 5 such as Linux and SunOS.

6
 7 51.Diff works by comparing entire lines for exact similarity. It works
 8 best when the lines in the file have not been re-ordered. If the re-ordering of lines
 9 is extensive, diff gets lost and fails to match even identical lines.

10 52.Graphical versions of the diff utility have also been provided on
 11 various platforms. These commonly attempt to align matching lines of code and
 12 show changes by color highlighting. Some more modern versions also attempt to
 13 match sections of code that have been moved out of order. WinDiff is a graphical
 14 version of diff for Windows, published by Microsoft.

15 **E. Obfuscation: Defeating Diff by Obscuring Code**

16 53.Because the diff comparison algorithm is so well known and
 17 readily available, programmers who copy code and who want to hide the fact that
 18 the code was copied can easily apply a number of techniques to defeat the diff
 19 comparison algorithm.

- 20
 21 a. One approach is the globally replace names with an equivalent. With
 22 integrated development environments (IDEs), such as Coco and Unity,
 23 it is quick and easy to change or translate keywords and identifiers in
 24 the source code. For example, a variable name, such as, `hero_id`,
 25 could be changed in all the files in the program to, `heroID`. This can
 26 be done in a few seconds with no real change to the structure,

27
 28 ⁸ James W. Hunt and M. Douglas McIlroy (June 1976). "An Algorithm for Differential File Comparison". Computing Science Technical Report, Bell Laboratories 41.

1 operation, or maintainability of the program. However, this global
2 change and replace (e.g. “Find and Replace”->”Replace All” in
3 Microsoft Word) will effectively defeat the diff algorithm, which will
4 now report every occurrence as a line that is different (not detected as
5 copied). If similar simple changes are made to the main function,
6 procedure, variable, and object names throughout the program, very
7 few lines will appear as “copied.”
8

- 9 b. Another approach is to reorder the code so the sections work the same
10 but have been moved around to avoid side-by-side comparison.
- 11 c. Another approach is to add or remove comments. Comments are
12 portions of source code which are ignored by the compiler or
13 interpreter (thus having not functional impact on object code) but
14 which are human readable and provide information about the nearby
15 lines of code. If a comment is added to a line of the code, the entire
16 line (which could otherwise have exactly identical source code) will
17 not be identified as being the same. Similarly if a comment is
18 removed or modified, the line will not be identified as being the same.
- 19 d. Another approach is to rewrite lines of code using different but
20 equivalent constructs. This defeats individual line-by-line comparison
21 for these lines. For example, Lua uses `self` to refer to its self, while
22 C# uses `this`.
- 23 e. Another approach is to rewrite lines of code using different but
24 equivalent languages, for example, translating from Lua to C# or
25 JSON. This defeats individual line-by-line comparison for these lines.
26 For example, Lua does not require a semi-colon at the end of the line.
27
28

1 C# does require a semi-colon. Thus every line of source code could be
 2 identical, except for the semi-colon.

3 f. Another approach is to rewrite lines of code using different but
 4 equivalent frameworks.

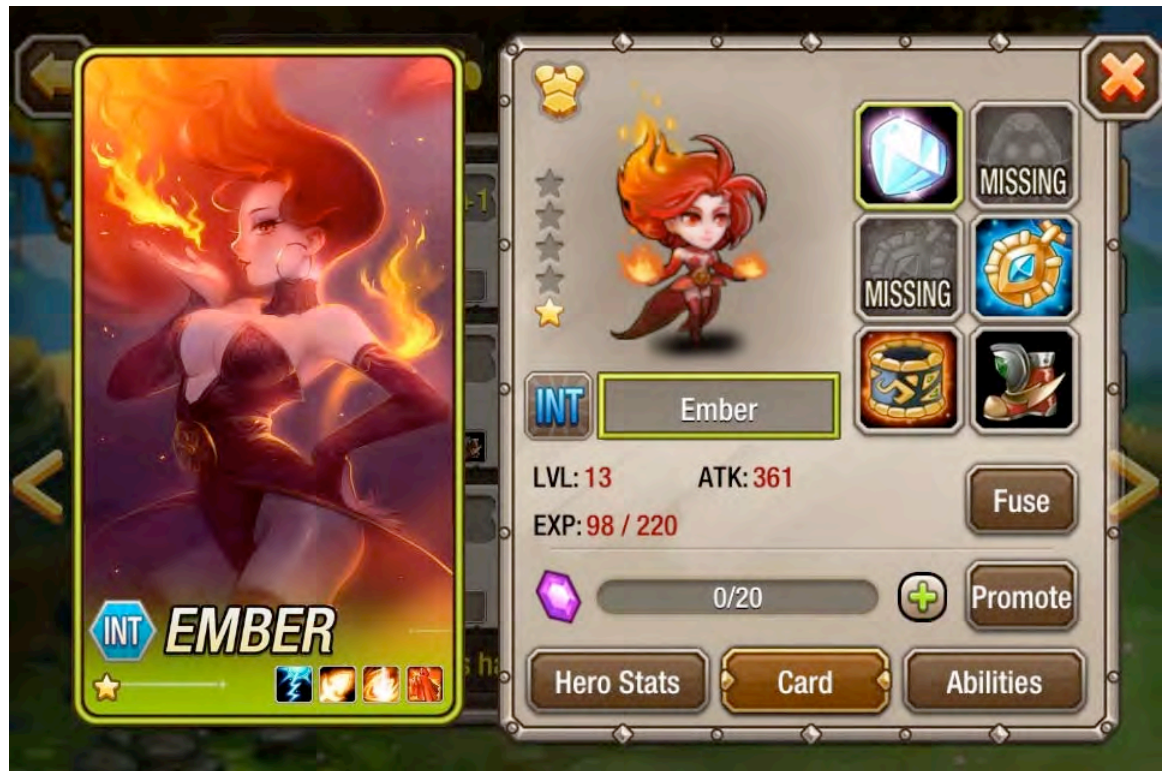
5 g. Another approach is to use a variable to hold an intermediate
 6 calculation or text conversion. For example a one line expression that
 7 performs a calculation, can be split into two or more lines where the
 8 first part of the calculation is performed separately with the result
 9 being saved in a variable on one line and then the rest of the
 10 expression being perform on a subsequent line using the variable in
 11 place of its corresponding calculation. This defeats the line-by-line
 12 comparison, but does not change underlying function.⁹

13 54. In my experience, use of these techniques to obscure similarities
 14 can indicate purposeful concealment of copying.

15 **X. OVERVIEW OF THE LILITH'S SWORD AND TOWER**

16 55. Sword and Tower has a hero character named Ember. In the
 17 source code she is referred to as "Fire Woman." (See below regarding Exhibit 5-4
 18 and Exhibit 5-7).

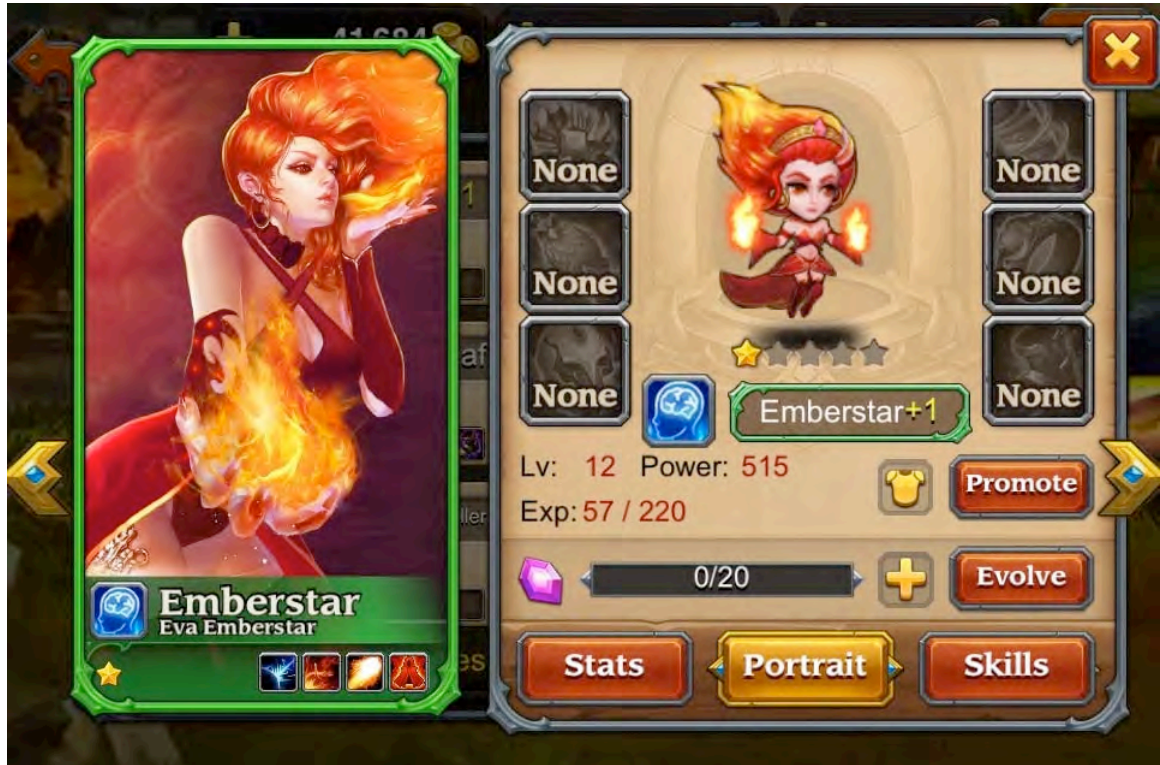
19
20
21
22
23
24
25
26
27
28 ⁹ In some cases such a change may not even change the object code, because a compiler generally
 breaks the expression down into the same sub-steps.



56. Five heroes are selected and go to battle in a battle scene against monsters. One monster is known as a “boss.”

XI. OVERVIEW OF THE UCOOL'S HEROES CHARGE

57. Heroes Charge has a hero character named Emberstar. In the source code she is also referred to as “Fire Woman.”



1 58. Also in Heros Charge, five heroes are selected and go to battle in a
2 battle scene against monsters. One monster is known as a “boss.”

3 **XII. ANALYSIS**

4 **A. Analysis Overview**

5 59. In my infringement analysis, I analyzed the computer software
6 programs, namely Plaintiff’s Sword and Tower program versus Defendant’s Heros
7 Charge program.

8 60. My analysis included looking at 1) literal copying, and 2) copying
9 of the non-literal elements of the original computer software programs.

10 61. In my analysis of literal copying, I also looked for source code that
11 was literally copied and obscured by multiple insignificant modifications. As
12 guided by Mitel, where literal copying is determined, I proceeded to determine if
13 the literal copying was protectable. I understand that certain elements are not
14 protectable because they constitute a method of operation under 17 U.S.C. §
15 102(b), they are unoriginal under 17 U.S.C. § 102(a), and they are dictated by
16 external factors and are thus unprotectable under the *scenes a faire* doctrine.

17 62. In my analysis of copying of the non-literal aspects of the
18 computer software programs, I applied the complete abstraction-filtration-
19 comparison test. I broke down the original programs into their constituent
20 structural parts and isolated each level of abstraction contained within that
21 structure. I further compared the original program with the corresponding
22 structural parts, at each level of abstraction (architectural level), of the copied work
23 to determine if Lilith’s source code has been copied.

24 **B. Analysis Results**

25 63. I found copying of both the literal and non-literal aspects of the
26 original computer software programs. Namely, I found that Plaintiff’s Sword and
27
28

1 Tower software program had been copied to Defendants' Heroes Charge software
2 program.

3 64. I prepared a number of exhibits to help shed light on the types of
4 literal and non-literal copying that I have found.

5 65. Exhibit 4-1 contains details regarding Plaintiff's Sword and Tower
6 source code files.¹⁰ These source code files are in the Lua programming language
7 as indicated by the ".lua" file extension.

8 66. Exhibit 4-2 contains details regarding Defendants' Heroes Charge
9 source code files. (UCOOL-SC-0000009 through UCOOL-SC-0000015) Note that
10 I was not provided the Lua version of Heroes Charge that was provided to Mr.
11 Kitchen. I was only provided the C# versions, as indicated by the ".cs" file
12 extension.

13 67. A comparison of the two lists shows that there are many files with
14 equivalent names. At the file level of abstraction, many of the same files are
15 organized into the same functional groups or classes (or "modules"). Further at the
16 component level of abstraction similar groups of files are found in similar
17 directories. For example a group of User Interface ("UI") files are found together
18 in both lists.

19 68. It is also clear that the Heroes Charge source code files are not a
20 complete set of source code files.

21 69. Exhibit 5 includes a number of side-by-side source code listings,
22 as well as single file listings. Sub-exhibits for each comparison are identified as
23 Exhibit 5-1, 5-2 and so forth.

24
25
26
27
28 ¹⁰ I reviewed several version of Lilith source code. Exhibit 4-1 is a list of the 1.9.3 version which
is the version Mr. Kitchen reviewed.

1 70. The side-by-side listings are consistent with exhibits that I have
 2 used over the last fifteen years to show source code similarities in many copyright,
 3 trade secret, and/or patent cases. In these sub-exhibits, the original Plaintiff's
 4 Sword and Tower files is shown on the left and the corresponding Defendants'
 5 Heroes Charge files are shown on the right. Lines of code that are literally
 6 identical, or literally translated, are shown in red and are underlined. Where
 7 identifiers were literally translated, a list of translation equivalents that were used
 8 is provided at the bottom of the text on the right hand side. Lines of code that are
 9 not literally identical, but are technically equivalent due to insignificant differences
 10 are shown in blue and are underlined. Lines shown in italics magenta and
 11 underlined are filtered.

13 71. When a literary work, such as a novel, is translated into another
 14 language, the translation is considered a derivative work and is protected by
 15 copyright. In the case of a novel, the translation into a natural language may not
 16 necessarily be a literal translation and as such may also be independently creative
 17 and expressive. Unlike natural language translation, the translation of source code
 18 identifiers here is a literal, word-for-word translation, and my opinion would not
 19 be independently creative and expressive, but is merely an effective means to
 20 obscure copying.

22 72. In both columns, line numbers show the original position in each
 23 respective file. The lines are lined up side-by-side with the corresponding lines
 24 where possible by inserting blank lines without line numbers. Long lines are
 25 wrapped within the respective columns as indicated by the absence of a line
 26 number and an arrow character that indicates line wrapping.

27 73. Exhibit 5-1 is a side-by-side listing of Sword and Tower's
 28 [REDACTED] source code file on the left and Heroes Charge's [REDACTED]

1 source code file on the right, as indicated in the column header on each page.
 2 (UCOOL-SC-0000007 through UCOOL-SC-0000008)

3 74. The only purpose of the [REDACTED] source code file is to
 4 define the various states that a hero, for example, might be in. The nine lines that
 5 are reddened and underlined have the exact same values, order, and syntax. The
 6 names have been obscured or obfuscated by changing the names of the identifiers.
 7 The Lilith names all have [REDACTED] as a prefix. The uCool uses the [REDACTED]
 8 [REDACTED] structure, which applies the concept to each of the matching lines. In
 9 my analysis of the two sets of code, I determined that “unit” has been translated to
 10 [REDACTED].¹¹ Thus, for example, [REDACTED] on the left is equivalent to the
 11 enumerated RoleState [REDACTED] on the right. The same is true for [REDACTED]
 12 [REDACTED]
 13 [REDACTED]
 14 [REDACTED] Statistically, 9 out of 24 lines were literally
 15 identical or translated, which is 37.5% of the entire file on the left. However, the 9
 16 lines which are copied represent the substantial purpose, and key content, of the
 17 file. Note also that blank lines are not counted as matches (in other words they are
 18 filtered). Thus, a score of 37.5% is a very high score under this methodology and
 19 represents substantial similarity.

20 75. Note that the changes in the identifier names would have caused
 21 the convention diff program to find no matching lines (i.e. a score of 0% matches).
 22 However, if both these files were converted to the same machine language, object
 23 code, or byte code, they would produce the same binary data and instructions.
 24 Thus, this is an example where the two sections of source code are technically
 25 equivalent due to insignificant differences but the similarity is obscured.
 26
 27
 28

¹¹ See also, e.g., Roman Exhibit 5-2, lines 104 on the left and 100 on the right.

1 76. Exhibit 5-2 is a side-by-side listing of Sword and Tower's
 2 [REDACTED] source code file on the left and Heroes Charge's
 3 [REDACTED] source code file on the right, as indicated in the column
 4 header on each page. (UCOOL-SC-00000016 through UCOOL-SC-0000054).
 5 Note that the difference in the names is merely a change from one standard naming
 6 convention to another standard naming convention. The all lower case name with
 7 underbar character is the original C naming convention. The use of uppercase
 8 letters to indicate a new word (instead of the underbar is a more modern
 9 convention, known as CamelCase, because it is like humps on a camel. However,
 10 it is clear that [REDACTED] has been translated to [REDACTED]¹²

12 77. Exhibit 5-2 is an example of the type of analysis that would be
 13 conducted during the litigation when the full set of source code is produced and
 14 when there is sufficient time to perform this level of analysis. I choose this one
 15 file as an example because [REDACTED] and its [REDACTED] function was
 16 discussed in the Kitchen Declaration at ¶159.

17 78. Exhibit 5-2 shows both literal copying and non-literal copying.
 18 The literal copying, for example as shown in both Exhibit 5-1 and Exhibit 5-2,
 19 suggests that uCool had access to Lilith's source code.

20 79. About 10 lines in Exhibit 5-2 are reddened and underlined
 21 indicating that those lines were literally identical, or literally translated.
 22 Additionally most of the lines in uCool's file on the right are either red or blue
 23 indicating a high degree of substantial similarity in the lines that were copied, as
 24 well as a very high degree of obscuration.
 25

26
 27
 28 ¹² Both Lilith and uCool use a mix of classic C and CamelCase naming. Neither has consistently applied one of the two standards.

1 80. At the flow chart or algorithm level of abstraction, the sequences
2 of blue lines with corresponding functions show that the same algorithm is being
3 implemented.

4 81. At the parameter list level of abstraction, the function calls show
5 identical and equivalent parameter lists.

6 82. At the function level of abstraction, many of the functions are
7 identical and show up in the same order inside the file.

8 83. Some of the functions are found in the right in a different order.
9 Moving entire functions to a different location in the file is insignificant in that it
10 has no impact on the operation of the computer software program. However,
11 moving some sections of the code to a different place in the source code makes it
12 harder for the casual observer, or the diff utility, to detect the existence of the
13 copied code. Because some sections of the code are out of sequence in the side-
14 by-side exhibits, I cannot show all of the code side by side; however, I have added
15 blank lines and marked the corresponding lines so that if the pages are held side by
16 side the similarity of the out of order sections can also be seen.

17 84. Because not all the corresponding code can be shown side-by-side,
18 when lines are marked, the corresponding lines show up elsewhere in the file.
19 Further some lines of the original source code were copied multiple times to the
20 Defendants' source code.

21 85. In addition to the literally identical, literally translated, or obscured
22 lines, I noticed a number of other indications that uCool's file was copied from
23 Lilith's file.

24 86. The identically name function, [REDACTED] starts on line
25 21 on the left and line 80 on the right. Line 23 on the left contains an identifier,
26
27
28

1 [REDACTED] which is matched on the right by a comment about fixing the
2 code to [REDACTED] [REDACTED]

3 87. The [REDACTED] function starts on line 157 on the left
4 and has been copied and obscured as the [REDACTED] function which starts on
5 line 117 on the right. On lines 124 and 125 on the right, calls to
6 [REDACTED] [REDACTED] have been commented
7 out. The existent of these equivalent function calls in the same sequence, which
8 are not used in the code, suggest that the code on the right was derived from the
9 code on the left.
10

11 88. The [REDACTED] function starts on line 382 on the left and has
12 almost entirely been copied and obscured as the [REDACTED] method which starts
13 on line 204 on the right. On line 210 on the right, a call to [REDACTED] has been
14 commented out.

15 89. The Kitchen Declaration at ¶159 states, “I have compared multiple
16 key functions in the various sets of code and found no such similarities. As just
17 one example, the [REDACTED] in the new Heroes Charge code contains a
18 local function [REDACTED]. I compared
19 this to the file FightingStage.lua (in Heroes Charge), which contains a local
20 function [REDACTED]. In addition to the
21 fact that the code is dissimilar between the two reset routines, the fact that the two
22 functions each receive a different number of input parameters (2 vs. 4) further
23 indicates that the C# code segments is not derivative of either the old Lua code for
24 Heroes Charge or the code for DOTA Legend.”
25

26 90. First, Exhibit 5-2 shows that Mr. Kitchen’s method was flawed
27 because numerous [REDACTED] [REDACTED] are very similar.
28

1 91. Second, Mr. Kitchen compared the wrong files. He compared

2 [REDACTED] [REDACTED] [REDACTED]
3 [REDACTED] [REDACTED].

4 92. Third, there are two reset functions defined in each file, namely

5 [REDACTED] [REDACTED] (on the left starting at lines 55 and 928, respectively) and
6 [REDACTED] [REDACTED] (on the right starting at lines 86 and 301, respectively). The
7
8 respective [REDACTED] functions call the respective [REDACTED] functions, at line 88 on
9 the left and line 91 on the right. Both the ultimately called [REDACTED] and
10 [REDACTED] are called with 2 equivalent parameters. Thus, showing the fallacy of
11 Mr. Kitchen's "different number of input parameters (2 vs. 4)" distinction and the
12 conclusion that "C# code segments is not derivative of either the old Lua code for
13 Heroes Charge or the code for DOTA Legend."

14 93. The Kitchen Declaration at ¶161 stated, "To compare the parties'
15 code, I identified unique and searchable elements within the C# source and did a
16 number of global searches for similar text within the entirety of the Heroes Charge
17 source directory. I was unable to identify any similar sections of code through this
18 technique."

19 94. Exhibit 5-2 shows many identical and similar strings showing the
20 flaw is Mr. Kitchen's methodology and the fallacy of his conclusion regarding
21 similar sections of code.

22 95. The Kitchen Declaration at ¶162 concludes, "In summary, in my
23 opinion, the new, Unity3d/C#-based source code for Heroes Charge shares no
24 expressive similarities with prior versions of the Heroes Charge or the DOTA
25 Legend Lua source code."

26 96. Exhibits 5-1 and 5-2 show that Mr. Kitchen summary opinion
27 regarding the new Unity3d/C#-based source is not correct.
28

1 97. Exhibit 5-3 provides a different type of side-by-side analysis.
 2 (UCOOL-SC-00000055 through UCOOL-SC-00000080) In this case found two
 3 uCool Unity3d/C#-based source that appeared to be an example of the type of
 4 changes being made. In Exhibit 5-3 the left hand side is [REDACTED], the
 5 same file on the right in Exhibit 5-3. On the right hand side of Exhibit 5-3 is
 6 [REDACTED] which appears to be derived from [REDACTED].
 7 As expected many lines are colored red. However there are some lines that are
 8 colored blue because they are not identical. For example the class name on line 10
 9 on the left and line 9 on the right is different. There are also some translations
 10 being made, for example on line 25 (left and right) [REDACTED] is translated to
 11 [REDACTED], which shows ongoing change to CamelCase.

12 98. On line 86 on the left [REDACTED] is modified, on line 124
 13 on the right to be [REDACTED]. However the newly added comment on line 122
 14 on the right still has the original [REDACTED]. As discussed above, this is
 15 passed as one of the two parameters to [REDACTED].

16 99. In its best light for the Defendants, Exhibit 5-3 shows that files that
 17 are clearly derived from one another have a high number of red or blue lines using
 18 my methodology, but may have many changes that are not highlighted or which
 19 occur naturally in the course of development. In its worst light for the Defendants,
 20 Exhibit 5-3 shows Defendants being caught in the act of obscuring copied code by
 21 changing identifiers such as [REDACTED] and [REDACTED] so that they
 22 would no longer be detected by conventional code comparison programs like diff.

23 100. In addition, I noticed a number of other indications that uCool's
 24 file was copied from Lilith's file.

25 101. Referring back to Exhibit 5-2 where I matched Lilith's
 26 [REDACTED] function with uCool's [REDACTED] function, uCool's
 27 [REDACTED]
 28 [REDACTED]

1 newly added comment on line 163 on the right in Exhibit 5-3, supports that
 2 matchup and the derivation with the comment [REDACTED] [REDACTED].

3 102. Exhibit 5-4 is a color print out of [REDACTED] (UCOOL-SC-
 4 00000081 through UCOOL-SC-00000125) showing the C# keywords as well as
 5 the large number of Chinese characters. In the preliminary review, I did not have
 6 time to do an in depth analysis of the Chinese characters. I learned many Chinese
 7 characters in the 1970s and have used this knowledge in my forensic work in other
 8 cases. For example, I have known the Chinese characters for gold and power for
 9 decades and easily recognize them in the source code and graphics.
 10

11 103. Based on the my analysis of the Chinese characters, I believe
 12 that during the course of the litigation the Chinese characters will shed further light
 13 indicating that the uCool source code is a derivative of the Lilith copyrighted
 14 source code.

15 104. Exhibit 5-4 shows several instances where tutorial strings are
 16 added. For example, on line 744, in the [REDACTED] [REDACTED]
 17 [REDACTED]
 18 [REDACTED] is added. Further, on line 799 in the [REDACTED]
 19 [REDACTED] [REDACTED] is added. I expect that these
 20 strings would be the kind that are loaded by the localization manager (see below
 21 regarding Exhibit 5-7). These type of strings would shed light on the question of
 22 copying locales, settings, characters, themes, plot, sequence of play, staging,
 23 playing of music and sound, and tutorial, as expressed in Lilith's Lua source code.
 24

25 105. Exhibit 5-5 is a side-by-side listing of Sword and Tower's
 26 [REDACTED] source code file on the left and Heroes Charge's
 27 [REDACTED] source code file on the right, as indicated in the column
 28 header. (UCOOL-SC-00000127). This is a one page excerpt, which has not be

1 highlighted, sheds light on additional substantial similarity and obscuration. For
 2 example consider the following character strings:

3 [REDACTED] [REDACTED] [REDACTED]
 4 [REDACTED] [REDACTED] [REDACTED]
 5 [REDACTED] [REDACTED] [REDACTED]

6 106. The difference between the two strings is that [REDACTED] has
 7 been translated to [REDACTED]

8 [REDACTED]

9 [REDACTED] Thus the difference is insignificant. However, there is an
 10 important similarity. Both contain the same misspelling of “tavern” as “tarven”.
 11 In prior cases, the Courts have found that copying of misspelled words is strongly
 12 indicative of copying.

13 107. Further, lines 314 through 321 contain other “effect” strings,
 14 some of which have the same misspellings. The related to common elements
 15 found in both the Lilith and the uCool games: bronze chest, silver chest, gold
 16 chest, magic soul, star shop, etc.

17 108. Exhibit 5-6 is the first page of a single file listing for uCool’s
 18 [REDACTED] [REDACTED] (UCOOL-SC-00000127). Line 27 calls
 19 [REDACTED] [REDACTED]. Line 28 converts the data to software objects
 20 [REDACTED] [REDACTED]. This indicates that the [REDACTED] is a source file written
 21 in JSON. The [REDACTED] source code has not been produced. It is likely that the
 22 data from many of Lilith’s Lua source code files have been translated into JSON.
 23

24 109. It is my opinion that uCool should have known that the
 25 Assets.txt file was a key JSON source code file that should have been produced
 26 and that it is related to the question of copying locales, settings, characters, themes,
 27 plot, sequence of play, staging, playing of music and sound, and tutorial, as
 28 expressed in Lilith’s Lua source code.

1 110. Exhibit 5-7 is a two-page excerpt of a single file listing for
 2 [REDACTED] [REDACTED] [REDACTED]. (UCOOL-SC-00000129 through
 3 UCOOL-SC-00000130). Lines 12 through 36 shows the codes for various natural
 4 languages include U.S. English (“en-US”) and Chinese (“zh-CH”). The
 5 constructor defined on line 39 calls Init on line 41. Starting on line 65, Init gets the
 6 the user’s language and language preferences from the Player Preferences and gets
 7 the system language (i.e. the language being displayed on the mobile phone). On
 8 line 73 it calls LoadDict which loads the localization strings from the
 9 Data/Language directory.
 10

11 111. It is my opinion that uCool should have known that the
 12 localization language files are key source code files that should have been
 13 produced and that they is related to the question of copying locales, settings,
 14 characters, themes, plot, sequence of play, staging, playing of music and sound,
 15 and tutorial, as expressed in Lilith’s Lua source code.

16 112. Exhibit 5-8 a single file listing of Lilith’s [REDACTED]
 17 which provides Lilith’s tutorial. It references [REDACTED] [REDACTED] and [REDACTED]
 18 [REDACTED] [REDACTED].
 19

20 113. The Lilith tutorial teaches a new user how to play the game. It
 21 has a specific sequence based on keys. Although the game is interactive and the
 22 user can make selections that affect the configuration, the tutorial is like a play
 23 with certain scenes that are presented on the stage in a certain order.

24 114. Exhibit 5-9 a single file listing of Lilith’s
 25 [REDACTED] which is Lilith’s tutorial maker. It references
 26 [REDACTED] on line 3.

27 115. Exhibit 5-10 a single file listing of Lilith’s
 28 [REDACTED] which includes the resources for Lilith’s tutorial. It

1 includes strings. For example, on line 312, in [REDACTED]
 2 [REDACTED]
 3 [REDACTED]
 4 [REDACTED]
 5 [REDACTED]
 6 [REDACTED] While I have not completed a full analysis of the respective
 7 tutorial files, nor a full analysis of the Chinese, these example show that there is
 8 substantial similarity between the Lilith Lua source code and the uCool C# code
 9 and in the localization files that have not been produced but which are evidenced
 10 in the uCool source code, for example as discussed above regarding Exhibit 5-4
 11 and Exhibit 5-7.

12 116. In this case the selection, functional breakdown, sequence,
 13 comments, and strings collectively are an example of expression that is original, is
 14 not dictated by external factors, and is not a mode of operation.

15 **XIII. CONCLUSION**

16 117. My primary opinions are summarized and stated in the
 17 Summary of Opinions Section above, and other opinions are stated throughout this
 18 declaration, including attachments.

19 118. I intend to conduct further analysis of information produced
 20 during discovery. I may also conduct research, analysis, or experiments that may
 21 result in new information. Specifically, I intend to review any additional expert
 22 reports or deposition testimony provided by Defendants and to respond to any
 23 arguments provided by Defendants or guidance provided by the Court. I reserve
 24 the right to supplement this declaration with the results of my continuing analysis.
 25 I also expect to be asked to comment upon the expert reports and testimony of
 26 others that has been or will be given in this matter.
 27
 28

1 119. Additionally, I intend to develop a number of graphics,
2 animations, and other videos for presentation at trial that will help the trier of fact
3 understand the structure and operation of the software and source code. I
4 understand that this material will be produced as demonstratives prior to any
5 hearings or trial in which they may be useful.
6

7 120. The findings and opinions set forth in this report are based on
8 my work and examinations to date. I may continue my examinations. I may also
9 receive additional documentation and other factual evidence over the course of this
10 litigation that will allow me to supplement and/or refine my opinions. I reserve the
11 right to add to, alter, or delete my opinions upon discovery of any additional
12 information. I reserve the right to make such changes as may be deemed
13 necessary.
14

15 121. This declaration was executed on July 30, 2015 at Sunnyvale,
16 California. I declare the foregoing under penalty of perjury under the laws of the
17 United States.
18

19 
Kendyl A. Román